Feb 29, 2024
Suyoung Park

〈고속 푸리에 변환 (Fast Fourier Transformation, FFT)〉

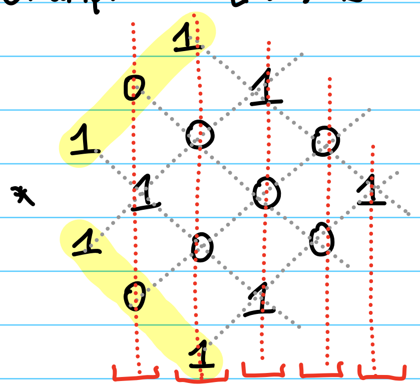
※ 참조 : invrtd.h tistory blog

벡터의 이산 푸리에 변환을 $O(n\log n)$으로 계산하는 알고리즘.

두 벡터의 합성곱 (convolution) 정의

let $c = a * b$ where $c_k = \sum_{i+j=k} a_i b_j$

Example $a = [1, 0, 1]$ $b = [1, 0, 1]$



$a \cdot b = [1, 2, 2, 0, 1]$

→ Time Complexity = O(len(a) · len(b))

⇒ 합성곱을 더 빠르게 계산하는 방법?

1. 각 벡터를 "다항식화" 하기.

   ex) $a = [1, 0, 1]$

   $\Rightarrow f(x) = 1 + 0x + 1x^2$

2. a를 f(x)로, b를 g(x)로 가정하면, 두 합성곱 c의 h(x)를 알 수 있지 않을까.

   ex) $f(0) \cdot g(0) = h(0)$, $x \in \mathbb{C}$ — 복소수 영역에서도 성립

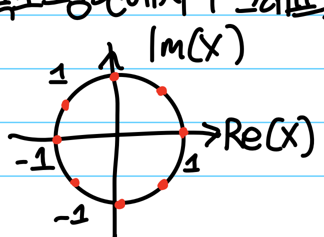
3. 즉, f가 주어지면 $[x_0, x_1, x_2]$를 $[f(x_0), f(x_1), f(x_2)]$로, 그 반대도 빠르게 푸는 문제로 간추려진다.

IDEA: $x_0, x_1, \cdots$ 들을, 1의 거듭제곱근으로 잡기

1의 n 제곱근 정의)

$$\omega_k = \cos\left(\frac{2\pi k}{n}\right) + i\sin\left(\frac{2\pi k}{n}\right),\ k = 0, \cdots, n-1$$

복소평면에서의 그래프)



문제점) FFT를 활용하기 위해서는 항상 n이 2의 거듭제곱이어야한다.
해결방안) 값이 0인 항들을 추가시켜 강제로 2의 거듭제곱으로 맞춰주면 된다!

$f(w_0)$, $f(w_1)$을 일일이 계산해서는 안된다 ⇒ MATRIX FORM으로 변환
ex) $n=8$

$$\begin{bmatrix} w_0 & w_0 & w_0 & w_0 & w_0 & w_0 & w_0 & w_0 \\ w_0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ w_0 & w_2 & w_4 & w_6 & w_0 & w_2 & w_4 & w_6 \\ w_0 & w_3 & w_6 & w_1 & w_4 & w_7 & w_2 & w_5 \\ w_0 & w_4 & w_1 & w_5 & w_2 & w_6 & w_3 & w_7 \\ w_0 & w_5 & w_2 & w_7 & w_4 & w_1 & w_6 & w_3 \\ w_0 & w_6 & w_4 & w_2 & w_0 & w_6 & w_4 & w_2 \\ w_0 & w_7 & w_6 & w_5 & w_4 & w_3 & w_2 & w_1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}$$

- 짝수 첨자와 홀수 첨자로 정렬하기

$$\left[\begin{array}{cccc:cccc} w_0 & w_0 & w_0 & w_0 & w_0 & w_0 & w_0 & w_0 \\ w_0 & w_2 & w_4 & w_6 & w_0 & w_2 & w_4 & w_6 \\ w_0 & w_4 & w_1 & w_5 & w_2 & w_6 & w_3 & w_7 \\ w_0 & w_6 & w_4 & w_2 & w_0 & w_6 & w_4 & w_2 \\ \hdashline w_0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ w_0 & w_3 & w_6 & w_1 & w_4 & w_7 & w_2 & w_5 \\ w_0 & w_5 & w_2 & w_7 & w_4 & w_1 & w_6 & w_3 \\ w_0 & w_7 & w_6 & w_5 & w_4 & w_3 & w_2 & w_1 \end{array}\right] \begin{bmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \\ \hdashline a_1 \\ a_3 \\ a_5 \\ a_7 \end{bmatrix}$$

- 점선으로 가르기

$$F(A) = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} B \\ C \end{bmatrix}$$

→ 놀랍게도 W_4이다!
→ W_4와 특수 행렬의 곱으로 표현 가능하다!

where $B = \begin{bmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \end{bmatrix}$, $C = \begin{bmatrix} a_1 \\ a_3 \\ a_5 \\ a_7 \end{bmatrix}$

- 적용

$$F(A) = \begin{bmatrix} W_4 & DW_4 \\ W_4 & -DW_4 \end{bmatrix} \begin{bmatrix} B \\ C \end{bmatrix} \qquad D = \begin{bmatrix} w_0 & 0 & 0 & 0 \\ 0 & w_1 & 0 & 0 \\ 0 & 0 & w_2 & 0 \\ 0 & 0 & 0 & w_3 \end{bmatrix}$$

$$\therefore F(A) = \begin{bmatrix} W_4 B & DW_4 C \\ W_4 B & -DW_4 C \end{bmatrix}$$

결국 $W_4(X)$도 $F(X)$로 치환할수 있기 때문에,

$$\boxed{F(A) = \begin{bmatrix} F(B) & DF(C) \\ F(B) & -DF(C) \end{bmatrix}}$$ → DIVIDE-CONQUER 사용가능!?

⇒ FFT 알고리즘 구현의 정수

가 성립!

- 역변환은 어떻게? (IFFT)

$w_0 \rightarrow f(w_0)$ 뿐만 아니라, $f(w_0) \rightarrow w_0$도 가능해야 한다.

→ 모든 성분을 역수로 취해주고, $\frac{1}{n}$을 곱하면 역행렬이 된다.

즉, FFT 구현에서 w_i를 $\frac{1}{w_i}$로 바꿔주고, 결과를 n으로 나누자.